What does QueryThreadCycleTime actually count?

April 29, 2016

Raymond Chen

A customer wanted some clarification on the `QueryThreadCycleTime` function.

> Based on the name, it suggests that it counts CPU cycles, but they observed that when the power profile is set to Balanced (which lets the CPU run at low frequencies when there is low demand), the thread cycle time increases twice as fast as when the power profile is set to High Performance (which sets the CPU to run at full speed all the time), even though the same code is executing in both scenarios. If it's the same code, shouldn't the cycle times be comparable? The function appears to be measuring elapsed time, not cycle time.

The documentation for the `QueryThreadCycleTime` function function contains this caveat:

> Do not attempt to convert the CPU clock cycles returned by **QueryThreadCycleTime** to elapsed time. This function uses timer services provided by the CPU, which can vary in implementation. For example, some CPUs will vary the frequency of the timer when changing the frequency at which the CPU runs and others will leave it at a fixed rate. The behavior of each CPU is described in the documentation provided by the CPU vendor.

This function uses the CPU's cycle counter instruction. For x86 and x64 processors, this means `RDTSC`. And according to Wikipedia, recent Intel processors fall into the second category, where the time stamp counter runs at a fixed rate regardless of CPU speed. This means that when the CPU is running at reduced speed, it can execute fewer instructions per counter. From the point of view of the program, it looks like the time stamp counter is running at a higher rate.

So what does `QueryThreadCycleTime` actually count?

The answer is "Whatever the CPU wants."

Raymond Chen

Follow